Historical data bases as a field for structured query language

Pasleau, Suzy

Historical Social Research, Vol. 14 — 1989 — No. 3, 23-29

Historical Data Bases as a Field for Structured Query Language

Suzy Pasleau *

I. From the Past to the Present

Since the historians began to make use of computer science in the sixties three revolutions have changed this new world in which they entered: the possibility to write softwares in performed programming languages, the irruption of micro-computer and the appearance of a new generation of softwares: the systems of maintenance of relational data bases. They exceed and often supplant the usual programming methods and languages. At first, they were only used for mainframes, they can now be adapted on microcomputers. The used of the standard sotware, that is to say a software which can be adapted on micro, mini and big mainframes facilitates the interconnection of mixed networks by using the same interfaces between the users and the material on the one hand, the material and the software on the other hand.

Up to now the enrichment possibilities of the files arranged to lend themselves to data processing through association or combination with other variables were not studied because each file was consulted separately without making use of the links it could have had with other analog files. The relational dimension does more than remedy this fragmentation. First it is a permanent invitation to the so-called secondary analysis: the latter retrieves the information which is included in a file but has not been studied yet. Then, making definite progress, the system of relational maintenance permits to locate and then to bring to the fore the relations between the variables.

This way of working is based on the fact that files which are subjected to certain constraints can be considered as representing a mathematical relation. Thus, they can be subjected to the theory of relations in order to solve the problems linked with the data manipulation in such files (1). The historians, who were glad to be able to solve case by case the problems noted during the exploitation of one or two series of variables can now see directly the advantages of such a system.

* Address all communications to Suzy Pasleau, Université de Liège, 32, Place du 20. Août, B-4000 Liège, Belgium.

Notwithstanding these improvements, one of the most difficult operations is the transition from the historical source to an input medium. The risk of such transfer is all the greater as it it necessary to include all the data and all the information that each source has in a relational system.

This requirement forces one to call into question the traditional coding procedure. The main difficulties come from the fact that the acquisition techniques must be adapted to the particularities of an old documentation and must be more user-friendly and able to the included in any kind of configuration.

More than a mere technical obligation, there is a challenge to take up. It will not be possible with a traditional type of software (2). Only a program generator is able to face the numerous situations we meet with historical sources. HDB/SQL - Historical data bases as a field for structured query languages - has achieved this double aim. Let us now describe to of its modules.

II. A Program Generator at the Service of the Historians

HDB/SQL can be used on every IBM PC or compatible, with a memory of 256 Kb. It is written in Turbo Pascal, and composed in two modules: the applications generator called SAISIE, the execution module, which allows the same applications to work independently from the creation module. In its complete version (applications generator and execution module), the software is located on one floppy disk. The user can copy the program on a hard disk.

II.1. How can the Acquisition Mask be Generated or how can we Define the Structure?

Before registering the variables on an input medium, the user must first build the structure of his future data base. This includes two stages. The first one consists in establishing the relations (3) within the data base, the latter being made of only one file where there is only one relation. The second one consists in preparing a list of the items and the way of acquisition of each relation. When this work is done, we can begin the data acquisition, on base of the defined structure.

The program which creates the acquisition masks and the corresponding files works in an interactive way that is to say, like a dialogue between the computer and the user. There is no need to know data processing at this moment.

After the operator has given a name to the data acquisition and to the first relation, the program asks him to define the fields. For each one it chooses:

a) the name: it is the title
b) the type: let us note the possibility to define variable zones.

So, HDB/SQL does not lose so much place during data recording. This option is as much advantageous as the data base is rich.

c) the length: It is the maximum number of characters it can contain.
d) the formal control: only the characters defined at the beginning can be registered.
e) the registration message

This one appears below the screen, during the registration of the item. It points out to the operator the way to complete the rubric.

When all the methods of a field acquisition are defined, the program passes to the definition of the following field. When the list of rubrics is closed - there is no defined limit - the operator may create an index on a field (except the fields like LCHAR) in order to accelerate processing.

A second relation can be created in the data base. Here we find one the main winning card of HDB/SQL, that is the possibility to generate many relations within the same data base. Let us take for example the militia registers (the data of which we would like to divide into three files).

The first file (A) groups the variables concerning the militiamen, chosen as basic unity.
The second file (B) contains the information concerning the family of the militiaman.
The third file (C) contains the data concerning the location of the militiaman.
After having arranged the three files A, B, and C, we must now fill them and there are two possible ways to do so.
The first one consists in using a classical system of relational data base (without any programming).

After the creation of the three files A, B, C, there are two possible opportunities: the first one is to register all the militiamen in the file A; then, after this to register their familiy in the file B and finally the data acquisition linked to the localization in the file C. In this case, we analyze the original record three times, and there are three times more risks of errors.

The second one consist in recording the first militiaman in the file A, to close the file A, to open the file B, to register the family, to close C, to open A... and so on, with the loss of time you well know.

It is not necessary to remain with these disadvantages. Let us add the problem of the hinge-element keeping the relation between the three files: either we create it on the basis of the registration number, or we choose

one (or some) fields and we repeat it (them) in the two other files. You have to choose, write and then repeat the hinge-element.

The second may invites us to use HDB/SQL.

The user generates three relations within the data base and, during the acquisition, he goes from one file to the other without any difficulties. Let us also precise that the hinge-element (see flow chart) is provided by the software: it establishes itself the link between the three files.

We can create as many relations as we want. The program shows in fine the orders for the creation of the tables (or relations) in a system of relational data acquisition.

II.2. The Data Acquisition

The ACQUISITION program conceives the data base from the structure defined by the STRUCT program.

At the beginning of each acquisition session a question appears on the screen, asking the user if he has already saved his data in a relational data base system or if it is the first utilization of the structure.

In case of affirmative answer, the program numbers the recordings, beginning with 1. If it is not the first utilization of the structure, the program reads the last recording which appears on the screen. If the user has not yet included and saved his data in a relational data base system, it is possible for him either to include new data in his file, or to destroy it and start again from recording n o 1 on.

With HDB/SQL the acquisition is accomplished with tables, following the modalities defined in the structure. For each table, the names of the fields that compose it appear. When the operator has typed the content of a field, the cursor is moved next to the title of the next field.

During the treatment of the next field, the message planned for the definition of the structure is indicated at the bottom of the screen.

After being recorded, the content of a field is submitted to the requested tests. If there is any mistake, the content of the item is deleted and the operator is obliged to reenter correct information.

After the field acquisition of the first table there are four possibilities for the operator:

- to continue the data acquisition in the same table, i.e. to add one or several new recordings. In this case, when the data base structure has been defined, either one table has been created, or we do not have any information to record for the first defined table (s);
- to correct the information in the recording which has just been made;
- to create a new recording in the same table and call up the data of the

preceding recording. This proceeding saves a lot of time when there are not many changes from one recording to another. It is impossible to realize it with a relational data base system without going back to the programming;

- to finish the data acquisition in this table. One can then begin the acquisition in another table or finish the session.

Once the acquisition is over, the program shows the operations to transfer and saves the recordings in a system of relational data bases. It is certainly one of the main card of HDB/SQL: it is as a link module with a relational data base system supporting the SQL language. We shall retain SQL WORKSTATION, SQL INFORMIX, MICRO INGRES and MICRO-ORACLE on micro computer; SQL DB/2, ORACLE, INGRES and IN-FORMIX on mainframe.

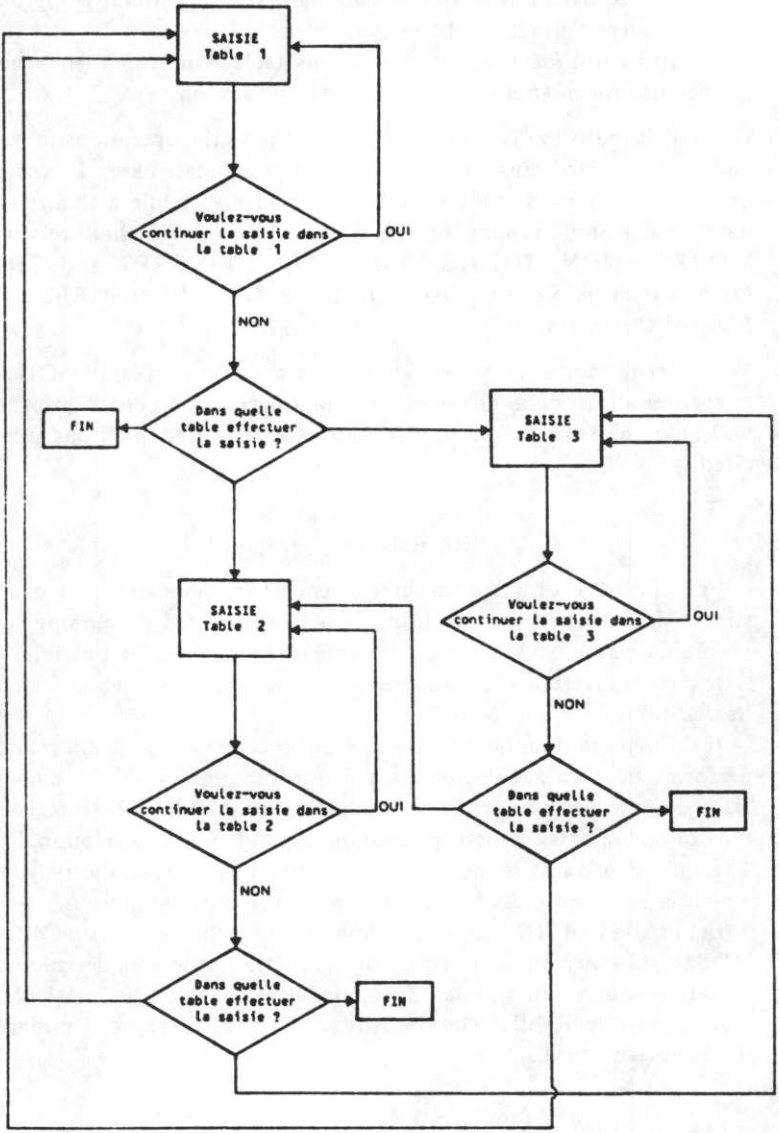
Whatever the software, whatever the type of computer, HDB/SQL is able to register all the data. The questions and the treatment of the information will be made from the relational data base software which has just been cited (4).

111. Balance of a Stage

For many historians, computer science long appeared as a real constraint. It must be said that learning a language and developing specific programs was mostly reserved for them. But today it is possible to use strong computerized and statistical means without any deep knowledge of its discipline.

If the exploitations do not pose any technical problem, it still remains to perform the data acquisition in a way compatible with the data bank. HDB/SQL makes this possible. The historians shall be able to register any type of documents, without programming, and this, in a relational mode. The only constraint is the use of a relational data base the language of which is SQL and these systems are more and more numerous.

Let us add that HDB/SQL is compatible with the new version dBase IV. HDB/SQL which finds its origin in acquisition programs designed of nominative sources in general, for population registers in particular (5), thanks to its flexibilities, conviviability and its facility, specially adapted to the historian's needs.



Flow chart of the procedure of militia register acquisition

Historical Social Research, Vol. 14 — 1989 — No. 3, 23-29

Notes

(1) S. Pasleau, »A propos des bases de données relationelles«, in Revue, Informatique et Statistique dans les Sciences Humaines, Liège, C.I.P.L., 1987, pp. 111-134.

(2) S. Pasleau, Legia II. La gestion, automatique des données en histoire, Laboratoire d'Informatique Documentaire et d'Histoire Quantitative, Liège, C.I.P.L., 1987, 110 pp.

(3) **A** relation is a two-dimensional table in which the columns are called »attributes« and the lines »t-uples«. Each line contains t attributes.

(4) These are precisely described in a book: S. Pasleau, »SQL. Langages et logiciels d'application«, Paris, Editests, 1988, 320 pp.

(5) S. Pasleau, Legia II, op. cit.